

第一题

关于密码的字符

- 密码的所有字符，是任意的字符，可能千奇百怪，包罗万象
- 我们要处理的字符，只是字母表里的大写和小写字母，所以只有 $26+26$ 个，是我们需要处理的
- 所有的小写字母，都转换成大写字母，再处理，不管是栅栏密码，还是凯撒密码

看看你没见过的测试样例吧

|JAAUFGHAGHWUIHNRWEEUHEEQIISHEA000JHYTUQWSWRBIHQHDEOHVUJWIOFHIWQSK7

'Fex欵俚mnd欵俩w欵俩kchzuph欵併q欵俩hah欵俛gj欵俛awii;欵俩ng欵脩ot欵修欵佻
kwuqpmfs,欵倅ikq欵修zki;欵併dw欵徬gcu欵脩a欵倅gmqg;欵偌eeq欵倬qqwgt欵倬lvw欵儻
am欵俶avc;欵偲yt欵俛kdwv欵併u欵俩uj,欵俶zv欵俶ab欵倬e欵併hhu欵俸ttg欵修pjk欵僂
qfsxmku' r欵偲xfafktku欵俠lwq欵俶r,欵俶vb欵俶ks欵佻xb欵俶cok欵俚udxe欵偲d欵偌w,
欵倅dn欵併qqo欵倬nqzouv欵俶f欵俵td欵俶yck' r欵脩nmtuk.欵伧ez,欵伧'xd欵僂qiax欵
俩ng欵徬l欵俵woshwh欵倅k欵俚nasqxokhup:欵俶qt欵俩oub欵俶dw欵俵uvc;欵脩vk欵佻xb
欵偌qwyl欵倬hxaffe欵俠m欵修o,欵俵d欵俶fan欵佻u欵偌d欵倝欵倬qo欵脩qipi.欵倻n欵
俶da欵俶vctk欵倬b欵佻tw欵修awp,欵俸es欵俛v欵脩wg欵倅gf欵俸i欵佻tw欵俵uzpeb欵俩
zt欵脩wg欵倅outrie.'A欵修ul欵脩e,欵脩nf欵脩q欵俶ktya欵脩zmdw欵併tir欵俶kr欵俚
yp欵脩pvkujlaz;欵俚tf欵俸fux欵俵欵俸qt欵偌ko欵倬pvw欵俶zv欵偲aqqu欵倝gq欵倬ldm
欵修tk欵俩wmzv欵俶z欵俩aj欵倝n欵俚koa欵儻v,欵俵欵偌dvfufbfqtuvx欵俸jxgvgj欵俶
ilpq欵偲hfp欵倅欵僂bx欵俛v欵脩acts:欵俶ew欵俸agm欵脩vmkqg,欵俚ettxgx;欵俛xh欵
俶adty' m欵俶qc欵俸xkkv,欵儻v欵俵pj欵偌q,欵俶fc欵伧欵俵cbv,欵倬fhfzkpm欵俚o欵儻
qmv欵俩sigdcs欵佻xb欵僂qqx欵俶dg欵偲gl'suve欵倅d.欵係tu欵俩jv欵倝rlt,欵偌tq欵俶
zuv欵俶qc欵倬tqoe:欵偲a欵俩lhuad欵俩zp欵脩yicdp欵倅qh欵俸notqa;欵俛fzdd欵俵
cuvhzw' p欵俶weilugqp.'Ohosct欵儻mp欵僂ojgn欵俶vo!'欵俵欵俸mac.欵?Kws欵脩bd欵脩
yht欵修欵偲aythdqj-欵脩ac欵倅ho欵儻uab欵偌欵俶ngcu-gdundd-欵俵mk欵儻qq欵俠fzg

- 第1行，我们还是认识的
- 第2行开始，就是我们看不懂的了，没关系，你只要记住，只处理26个大写字母和26个小写字母就可以了，那些不认识的字符就不会困扰你了！

怎么处理RE?

- RE基本都是越界造成的，可能有2个来源
 - 1是那些你不认识的字符，忽略他们就可以了
 - 2是凯撒密码的有效输出字符是A至Z，你处理凯撒密码时，要注意，加整数N后超过Z的字母是需要循环来到A的，这是一个圈，不能超出范围。
- 字符可以和整型数据一样，做四则运算
- 一个int类型的a，可以用char c = (char) a;这样的强制类型转换，转换成char类型

第二题

关于事件

- 共有3种事件类型（event type），分别是类型1、2、3
- 类型1的事件到来后，我们要做什么？
 - 存到stm中，留做后续处理；
 - 计算stm中每一个String的长度，将这些长度值排序，取长度值的中位数L，这其实就是counterStatements（）方法需要做的事情；
 - 将ai记为L，如果没有新的type1事件到来，则本轮ai值与上轮相等。
- 类型2的事件到来后，我们要做什么？
 - 存到tb中，留做后续处理；
 - 将这个新到达的事件，与stm中的已有String作比较，有一组相等则匹配对数加1，有几组相等则匹配对数增加几，这其实就是pairMatch（）方法需要做的事情；
 - 将bi记为当前已有匹配对数，如果没有新的type2事件到来，则本轮bi值与上轮相等；
 - 输出本轮的ai值。
- 类型3的事件到来后，我们要做什么？
 - 输出本轮的bi值

关于 I (Influence Gauge) 的计算

- 什么情况，会对 I 的值产生影响？
 - 类型2的事件达到， $I + 1$
 - 类型1的事件达到，且新计算出的 L 值比 I 大，则需要将 I 的值减去当前stm中String的个数，即stm的size
 - 以上2种操作就是alterInfluenceGauge () 方法需要做的事情

关于Score的计算

- 什么时候计算Score?
 - 所有事件到达后
- 怎么算才能避免超时?
 - 我们使用Score时，其实看的是它与0的关系，所以可能它具体等于多少，我们可以不用精确获得
 - 请仔细观察Score的计算公式，试着展开几项再合并下，你就会发现方法了
 - 令人惊喜的是，如果你找到了避免超时的方法，也就同时解决了下面的超long范围问题，你就不用考虑下面这一条问题了。
- 超时的测试点，已变为加分项，如果我不想处理这部分，还有什么需要注意的地方?
 - 请注意测试样例可能个数很多，长度很长，互相之间相乘再相加很容易超过long的取值范围，就overflow了，那就会得到很多Runtime Error了

最后的Fail和Qi Fei受什么影响

- 2处有影响
 - 一处是 I , 一旦 $I < 0$, 则直接Fail
 - 另一处是Score, $\text{Score} < 0$, 则Fail
 - 当 $\text{Score} \geq 0$ 且 $I \geq 0$, 则Qi Fei

几个小细节

- a, b和m都是长度为n的数组
- 每轮b的计算, 只考虑新达到的tb即可, 不用考虑前面已存在的tb的statement
- 类型1和类型2都长这样
 - int数据 (空格) String数据
- 只有类型3长这样
 - int数据 (其实就是一个数字3)
- 作业文档里提供的那些好复杂的代码, 干什么的? 必须用吗?
 - MaxHeap、MinHeap是排序用的, 可以不用, 你完全可以自己排序, 或者使用JAVA的各种排序方法, 自己定!
 - HashMap 是字符串匹配时用的, 可以不用, 用你自己钟爱的方法, 也可以!
 - 链接里提供的Qreader等其余的那些方法, 是快读快写的一个框架, 提高读写速度, 也完全可以不用, 用我们熟悉的Scanner和println, 都可以!

看看输入输出样例吧

Sample input 2

```
1 7 0
2 2 5 -3 -1 2 1 0
3 1 No
4 1 Hemust'veseenthekillertakethknife
5 1 ItwassodarkIcouldn'tseemyfood
6 1 Butitwassuperpitchblack
7 2 I
8 2 No
9 3
```

我是/

我是n

我们是m数组，一共有n个整数

从3行到9行，一共有n个事件

所有事件都到达后，输出Fail或Qi Fei，因为/在第5个事件时小于0了，所以最后输出Fail

Sample output 2

```
1 23
2 23
3 1
4 Fail
```

第7个事件：类型3，输出bi的值，所以输出1

第6个事件：类型2，所以把“No”加入到tb里面。比较“No”与stm当前所有String，有1个匹配的项，所以此处b值为1，所以b6为1。a6仍为23。类型2事件到来是，应输出ai的值，所以又输出了23。

第1个事件：类型1，所以把“No”加入到stm里面。“No”长度为2，因为stm里只有一个statement，所以长度的中位数就是2，所以a1的值为2

第2个事件：类型1，所以把

“Hemust'veseenthekillertakethknife”加入到stm里面。“Hemust'veseenthekillertakethknife”长度为34，现在stm里有2个statements，长度分别是2和34，排序后是2,34，所以a2的值为2

第3个事件：类型1，所以把

“ItwassodarkIcouldn'tseemyfood”加入到stm里面。“ItwassodarkIcouldn'tseemyfood”长度为29，现在stm里有3个statements，长度分别是2，34和29，排序后是2,29,34，所以a3的值为29

第4个事件：类型1，所以把“Butitwassuperpitchblack”加入到stm里面。“Butitwassuperpitchblack”长度为23，现在stm里有4个statements，长度分别是2，34，29和23，排序后是2,23,29,34，所以a4的值为23

第5个事件：类型2，所以把“I”加入到tb里面。同时/值加1，原来/值为0。当前L（也就是ai）为23，比/大，所以/值被破坏，需要减去stm内元素个数，即减4， $0+1-4=-3$ ，小于0了！比较“I”与stm当前所有String，没有匹配的项，所以此处b值为0，需要注意的是，此处的b的下标应为5，所以b5为0，而b1至b4为默认值0。那么a5呢？由于是类型2的事件，stm没有发生变化，所以a5值与a4值相等。此外，类型2事件到来时，应输出ai的值，所以我们输出了23。

再看看能引起TLE或RE的样例吧

我是n 我是/
5000 25
-437818513 -910632339 529424353 -637807227 -686609698 -995142165 -156529448
266436024 218266587 -770748998 407418969 -143637766 -515894067
68682955 -69336339 705082204 884203054 -180548474 -337598642 -422259635
714314511 -413586176 86923565 -599172683 135160597 859721332
669297460 -867793079 -580523465 -692207413 -413139672 -829586064 -882060360
94798747 674694841 -741045918 149122473 -98156952 -257658082 -357222237
498038311 351166158 -550496604 -175221433 -481349965
976555191 -357975042 -950010960 162199960 626708637 -376970249 232811307
505820725 -653864740 833718253 620568964 716933309 -553336232
531990991 -280372236 -764158722 -551560258 892107387 588588555 -667340686

- n值为5000，很大了，所以计算Score时很容易就会超过long，从而overflow了
- m的取值，也可以很大

篇幅有限，不一一截图展示了

同样还是那个样例

```
-  
2  
vtlnivlkrllzlopybtfmugdlpbccdfityxxvzmzbihszgelifbnpbothhkkparhagbdzvjrcepvgboco  
ulpazvqjyfmmthrmmxmcfg  
2  
opjgezmbaqjazzhuouwfoidyqtqjvzfwlfctvtsmhsrzxxnjismabovnytdaghyrujvocrljvrwlms  
tmtqfgevnrczsduvvivmnd  
1  
igyguqlsasmytlphrnomtfkjeunvgkiqydjzxybbmysokowiqdksxctuoazfxwgkmpuyjomfiaimesr  
rugdsvuwrgebtetptnqbzz  
2 qczltzggx  
2  
ruugznrifyrwoxaktzyovspzgganxdfwqkluaaahjvbqwopblizpvgezgyftjwwwsaancfxanytughc  
mpymtbfisgqrsfytyfwg  
3  
2 odurjwkqwsqqivmakrbqyctsvsuimeamuqrcgwqvdzcxpxxyjwirxftz  
1  
lchkavchybpucrrrhreczrrcchopeamriniexpglkxxxnzzrpvcsmjvghqnrghnznsxumtvdxrijh  
ahaoguokrqtlyhnffuvcpf  
3
```

- 每一个statement, 也可以比较长

我对于TLE和RE还是很气愤！

- 看完前面的内容，大家应该对题目的整个流程比较了解了，我再整理下思路：
 - 先判断事件类型，再作相应处理；
 - 增加对于 t 值的计算逻辑，想一想，应在哪个类型的事件里
 - 所有事件都达到后，再处理Score，结合前面 t 的情况，给出Fail或Qi Fei的输出
- 都做完后，还是有TLE和RE的错误！！
 - 那我们又回到了“关于Score的计算”那一页的问题了，还是把那个Score的计算公式展开看看吧，一定会发现的！
 - 并且，再次说一下，我们真的关心的，其实是Score与0的关系！